ACEEE Int. J. on Electrical and Power Engineering, Vol. 4, No. 2, Aug 2013

FPGA Optimized Fuzzy Controller Design for Magnetic Ball Levitation using Genetic Algorithms

Basil Hamed[1], Hosam Abu Elreesh[2]
Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine
[1]Email: bhamed@iugaza.edu
[2]Email: m_hossam@hotmail.com

***Abstract*— This paper presents an optimum approach for designing of fuzzy controller for nonlinear system using FPGA technology with Genetic Algorithms (GA) optimization tool. A magnetic levitation system is considered as a case study and the fuzzy controller is designed to keep a magnetic object suspended in the air counteracting the weight of the object. Fuzzy controller will be implemented using FPGA chip. Genetic Algorithm (GA) is used in this paper as optimization method that optimizes the membership, output gain and inputs gains of the fuzzy controllers. The design will use a high-level programming language HDL for implementing the fuzzy logic controller using the Xfuzzy tools to implement the fuzzy logic controller into HDL code. This paper, advocates a novel approach to implement the fuzzy logic controller for magnetic ball levitation system by using FPGA with GA.**

***Index Terms:* — Fuzzy Control, PI, FPGA, Genetic Algorithms Magnetic Levitation Ball, VHDL.**

I. Introduction

In the recent years Fuzzy controller is used to control complex engineering problems which are difficult to solve by classical methods. Finding many different hardware implementations of fuzzy logic systems (FLSs), general-purpose microprocessors and microcontrollers are mostly used for implementing FLS in hardware, but with the complex systems these devices cannot perform operations assigned to it as required. In recent years many studies emerged illustrate the different ways to implement fuzzy control using FPGA in different application. The advantage of using FPGA is suitable for fast implementation and quick hardware verification. FPGA based systems are flexible and can be reprogrammed unlimited number of times. J.E. Bonilla, V.H. Grisales and M.A. Melgarejo [1]; the fuzzy controller architecture in this paper focused on the treatment of errors and changes in errors with tuning gains. There are many ways of how to use GA in fuzzy control. The most extended GFS type is the genetic fuzzy rule-based system (GFRBS), where an EA is employed to learn or tune different components of an FRBS. The objective of a genetic tuning process is to adapt a given fuzzy rule set such that the resulting FRBS demonstrates better performance [2]. This paper presented the development of an FPGA-based proportional-differential (PD) fuzzy LUT controller. The fuzzy inference used a 256-value LUT. This method was used due to its reduced computation time cost. McKenna and Wilamowski [3] have investigated method to implement fuzzy logic controller (FLC) on a field FPGA and obtained very smooth control surfaces. Vuong et al [4]; described a methodology of implementing FLS using very high speed integrated circuit hardware description language (VHDL). The main advantages of using HDL are rapid prototyping and allowing usage of powerful synthesis tools such as Xilinx ISE, Synopsys, Mentor Graphic, or Cadence to be targeted easily and efficiently. S. Ravi and P. A. Balakrishnan [5] presented Genetic Algorithm based Fuzzy Logic Controller for temperature control in a plastic extrusion is developed and tested through a simulation study. A novel GA based FLC method is implemented to design a practicable advanced controller. Mohanad Alata, Mohammad Molhem and Khaled Al Masri [6] presented a solution of first, second, and third order systems, using the absolute average error as a fitness function, the genetic algorithm manipulate all parameters of the fuzzy controller to find the optimum solution.

II. Fuzzy Control

Fuzzy Control applies fuzzy logic to the control of processes by utilizing different categories, usually 'error' and 'change of error', for the process state and applying rules to decide a level of output. There are many models of FLC, but the most famous are the Mamdani model, Takagi- Sugeno-Kang (TSK) model and Kosko's additive model (SAM) [7]. This paper uses Mamdani model.

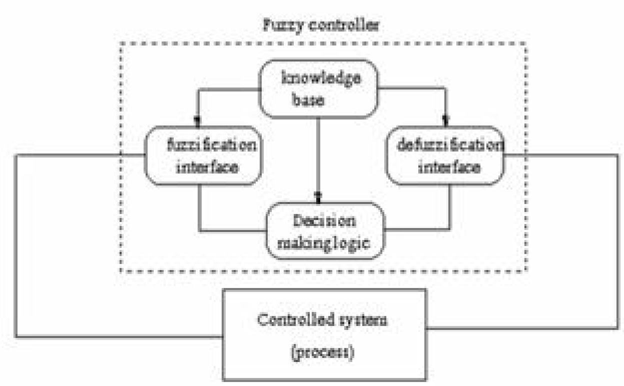


Figure 1: Mamdani Model

IF (X is A) and (Y is B) … THEN (Z is C).

Where A, and B are membership of the inputs, C is membership of the output as shown in Figure 1. Mamdani model block consist of three stages.

- **Fuzzification: -** Fuzzification means converting a crisp value of process variable into a fuzzy set. In order to make it compatible with the fuzzy set representation of

DOI: 01.IJEPE.4.2.1384

ACEEE Int. J. on Electrical and Power Engineering, Vol. 4, No. 2, Aug 2013

the process state variable.

- **Fuzzy Associative Memory (FAM): -** FAM is a set of fuzzy associations between the input and the output [6]. This stage consists of two parts:

A. Knowledge base

Knowledge base contains a data base and rule base. Data base provides necessary definitions for linguistic rules, and the rules base consist of the IF-THEN rules, which can be derived by using four ways:

1. Expert Experience and control engineering knowledge.
2. Based on fuzzy modeling of human operators central action.
3. Based on learning
4. Based on fuzzy model of a process.

B. Decision-Making

Decision-Making means choosing the most appropriate action from several possible actions.

- **Defuzzification: -** Defuzzification strategy is aimed at producing a non-fuzzy control action, or we can say defuzzification means the conversion of the fuzzy output values into crisp values.

III. FPGA

FPGA is digital integrated circuits (ICs) that have electronics blocks which can be programmed, and these blocks have configurable interconnection between them. These blocks can be used by the designed engineer to perform a huge range of tasks.

Xilinx has been designed first FPGA in 1984 but started being in use by the engineers in 1990 [8]. Every FPGA has three major components, configurable logic blocks (CLBs), input/output blocks (IOBs), and interconnects, Figure 2 shows theses blocks. CLBs are responsible for building the logical circuit for the user. IOBs are responsible for the interface between package pins and internal signal lines. Interconnects are responsible for routing paths to connect the inputs and outputs of the CLB and IOB [9].

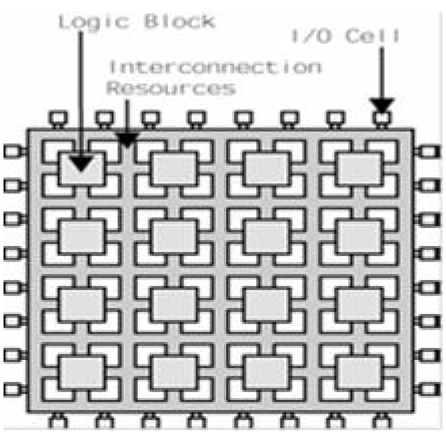


Figure 2: FPGA Architecture

These days, FPGAs offer the possibility of using dedicated blocks such as memories RAM, multipliers cabled PCI interfaces and processor cores. The design of control architectures is done using CAD tools. There are two commonly used languages, Very high speed integrated VHDL and Verilog. These two languages are standardized and compatible with all FPGA technologies previously introduced. Figure 3 shows the FPGA programming steps. This paper will use the Spartan 3e FPGA from Xilinx company.

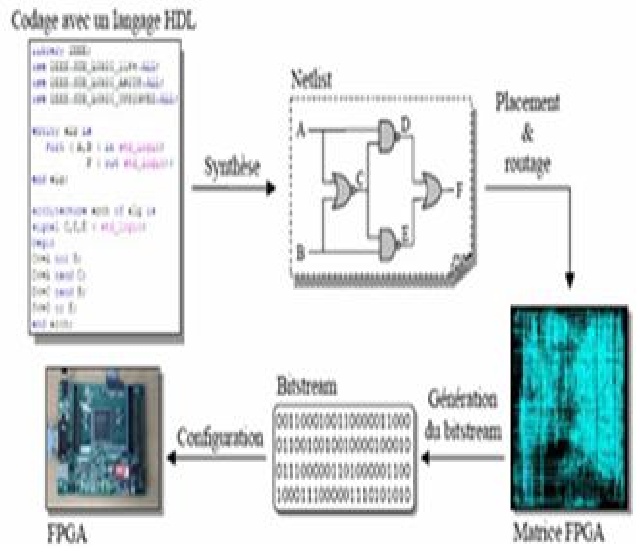


Figure 3: Programming a FPGA

IV. Genetic Algorithms

Genetic Algorithms are reliable and robust methods for searching solution spaces [10]. GA is general purpose search algorithm which uses principles inspired by neutral genetic to find solutions to problems [11][12] by using Survival of the fittest principle. The basic idea is to maintain a population of chromosomes, which represent candidate to the concrete problems that will be solve, through a process of computation and controlled variation. Each structure of chromosome in the population represent one of the possible solution of the problem and the fitness test of these chromosomes can determine which chromosomes are used to form a new chromosomes that will be used in computational process. As in natural the new chromosomes are created by some operations such as crossover and mutations. There is another operation which called reproduction. This operation is added to achieve the survival of the fittest principle. In recent years, GA is used in many applications specially in optimization and search problems and had a great measure of success; the main reason of this success that it can start from any solutions, and generate other solutions that converge to the optimal solution in less time versus other classical search tools (enumerative, heuristic). “GA is theoretically and empirically proven to provide a robust search in complex spaces; thereby offering a valid approach to problems requiring efficient and effective searches” [2]. Figure 3 shows the steps of GA.

DOI: 01.IJEPE.4.2.1384

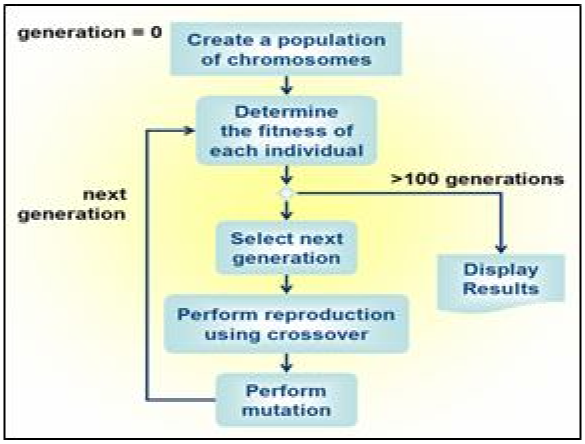


Figure 4: The basic Genetic Algorithm

1. Selection: In this process the developer will choose the pairs of parents that will be crossed. There are many methods can be used such as Roulette Wheel Selection. Rank Selection and Stochastic Universal Sampling [13].
2. Crossover: Crossover is the process that takes two parents of solutions and generates a new offspring. There are many methods can be used such as Single-Point Crossover (Figure 5), two-point crossover and uniform crossover [13].

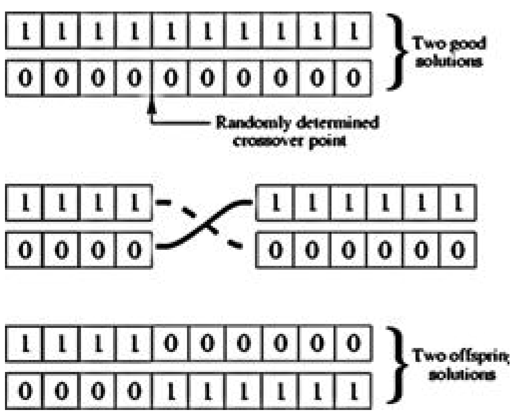


Figure 5: Single Point Crossover

3. Mutation: Mutation means swap one bit in binary coding or changes one number if the chromosome consists of numbers [13].

There are many optimization methods, but GA has some advantages over these methods such as [14]:
1- GA does not deal with data directly but works with encoded data.
2- GA uses least information such as fitness function to solve problems does not need derivation.
3- GA uses probability laws rather than certain laws.
4- GA generate populations of answer not just one answer.
5- Almost all conventional optimization techniques search from a single point but GA always operates on a whole population of points (parallelism).

In this paper the fuzzy membership inputs and output membership functions will be used as variables that will be optimized using GA. Every triangular membership has three variable can effect on the shape of it; so the each chromosome will has the number of genes every genes refer to one parameter that effect on the membership shape. For example every triangular membership can represents be three genes because it has three parameters that control of its shape (center edge, left edge and right edge). If the inputs of fuzzy controller have seven triangular memberships, then every chromosome of the population will have twenty one genes (7x3). The main problem in GA is how to choose fitness function. Minimize output error is one of importance aims in control systems. In control applications there are different fitness function that may be used [15].

1- $fitness\ .value = \int_0^\infty e^2(t)dt$ Sum of squared error (1)

Where (e) is the error signal, this function can track error quickly, but easily gives rise to oscillation.

2- $fitness.value = \int_0^\infty |e\ (t)|dt$ Sum of absolute error (2)

This function can obtain good response, but its selection performance is not good.

3- $fitness\ .value = \int_0^\infty te^2(t)dt$ Sum of time weighted squared error (3)

This function can gives fast tracking and good response. In this paper the sum of absolute value of error will be used as fitness function.

V. Simulation

The CE 152 Magnetic Levitation Model is one of the ranges of educational scale models offered by Humusoft Company for teaching system dynamics and control engineering principles. The model belongs to the range of teaching systems directly controllable by a PC computer in real time. The CE 152 Magnetic Levitation model is one dimensional strongly unstable system designed for studying system dynamics and experimenting with number of different control algorithms based on classical and control theory. Figure 6 shows CE 152 model [16].

The model shown in Figure 6 consists of the following blocks:
1. D/A converter.
2. Power amplifier.
3. Ball & coil subsystem.
4. Position sensor.
5. A/D converter.

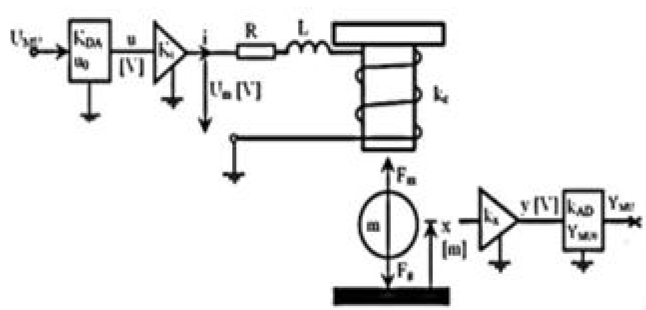


Figure 6: CE152 Magnetic Levitation Model

DOI: 01.IJEPE.4.2.1384

The mathematical equation of this model is:

$$\begin{bmatrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{i^2 k_c}{m_k (x_1 - x_0)^2} - g - \dfrac{k_{fv} x_2}{m_k} \\ \dot{x}_3 = \dot{i} \end{bmatrix} \quad \text{(nonlinear state space model)}$$

Where
x_1 ball position, x_2 ball velocity and x_3 coil current
m_k = ball mass [kg].
g = gravity acceleration constant [m.s^{-2}].
k_{fv} = viscose friction.
x_0 = coil bias [m].
K_c = coil constant.

A. Fuzzy Controller Design for CE152 Model

To apply the fuzzy logic controller to the magnetic levitation CE 152, certain properties of the system are exploited so that the design of the controller can be made easier. As the system is symmetrical, it is assumed that symmetrical membership functions about the y-axis will provide a valid controller. A symmetrical rule-base is also assumed. The fuzzy controller of magnetic levitation uses Mamdani model. The FLC has two inputs which are error and change of error and the output is the change of voltage. Figure 7 shows the membership functions of fuzzy controller using Fuzzy Toolbox of Matlab software. The ranges of the inputs and output is [-1 1]. All have 7 membership functions.

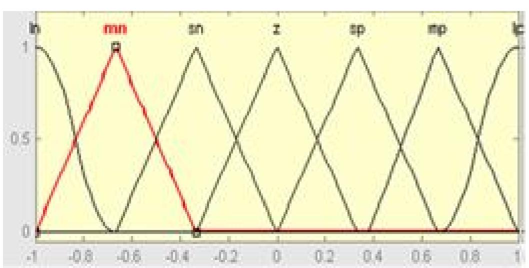


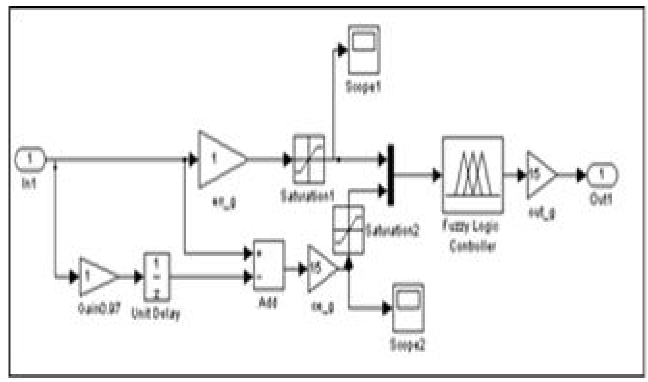
Figure 7: Memberships of Inputs and Output Fuzzy Controller

The next step is to write the rules of the fuzzy controller. These rules are chosen based on knowledge base and experts. Table 1 shows the rules base as a matrix After designing fuzzy controller, the controller will be connected to the CE152 magnetic levitation Matlab Simulink model. Figure 8 a, b and c shows the fuzzy controller, proportional integral controller (PI) subsystems and all system respectively. The integral part is used to convert the change of voltage value that come from fuzzy controller to real voltage which will act on the magnetic levitation CE152. Using tuning method the good proportional gain is 0.1 and the integral gain is 1.

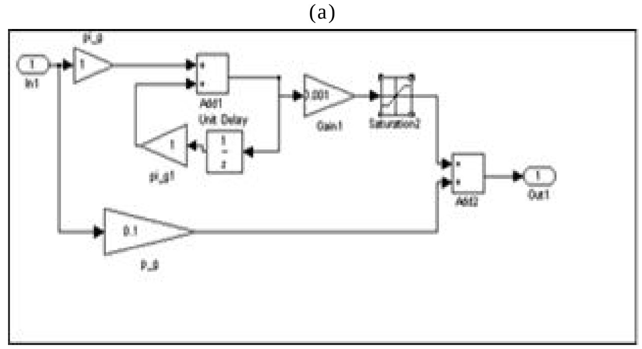
Figure 9 shows the output of the magnetic levitation after connecting to the fuzzy controller. The set point is unit step has value 0.5 which is in the center of the gap; this point is one of the equilibrium points of magnetic levitation CE152 model. There is no overshoot and the settling time is nearly 0.2 sec, rising time is 0.106 sec.

TABLE I. FUZZY CONTROL RULE BASE OF MAGNETIC LEVITATION

e / ce	ln	mn	sn	z	sp	mp	lp
ln	ln	ln	ln	ln	mn	sn	z
mn	ln	ln	ln	mn	sn	z	sp
sn	ln	ln	mn	sn	z	sp	mp
z	ln	mn	sn	z	sp	mp	lp
sp	mn	sn	z	sp	mp	lp	lp
mp	sn	z	sp	mp	lp	lp	lp
lp	z	sp	mp	lp	lp	lp	lp



(a)



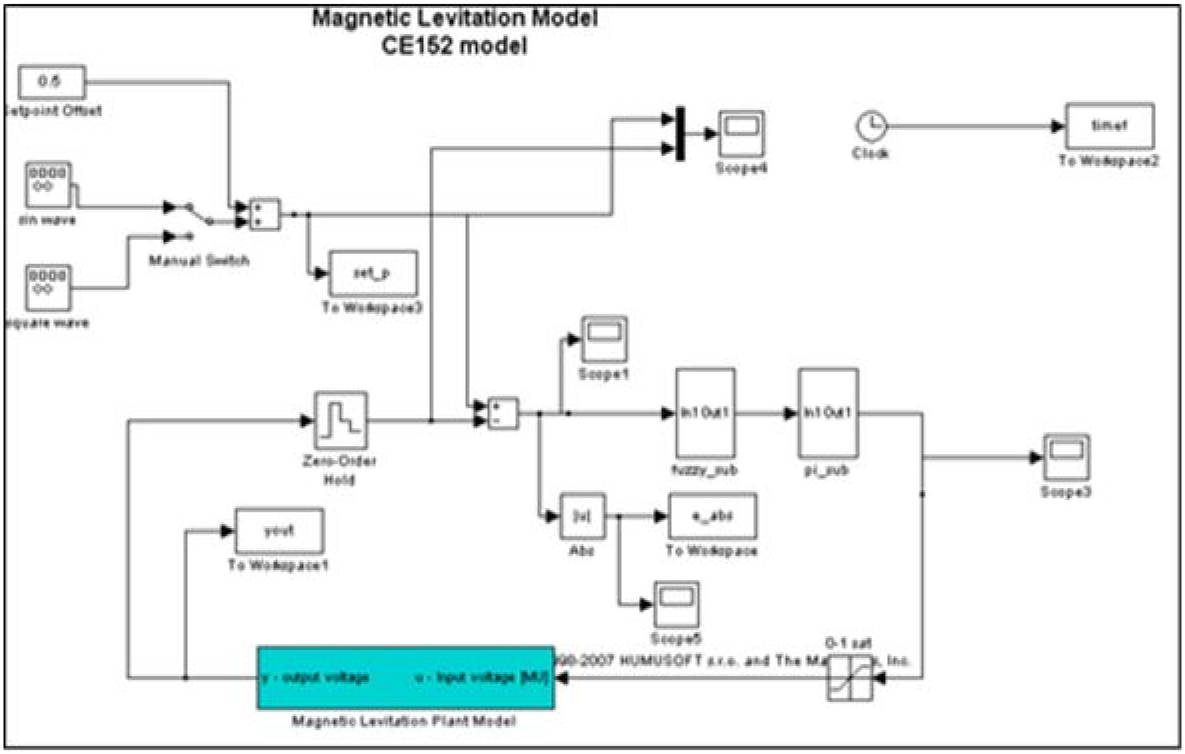
(b)

Figure 10 and Figure 11 show the output response with two additional set points. These figures show that the fuzzy controller can keep the stability of the system with various set points.

B. VHDL Fuzzy Controller Implementation

To program FPGA Xfuzzy tool is used. Xfuzzy is a CAD tool that was developed using JAVA language. This tool is a combination of several tools that covering the four stages of FLS design: description, tuning, verification, and synthesis stages. The controller consists of three main parts, ADC, FPGA, and DAC. ADC and DAC that will be used in this paper have 8 bits resolution. The integrated circuits (ICs) that will be used are PIC 16f877 as ADC, and DAC 0800.

DOI: 01.IJEPE.4.2.1384



(c)

Figure 8: a) Fuzzy Controller Subsystem - b) PI Subsystem –c) Magnetic Levitation Model

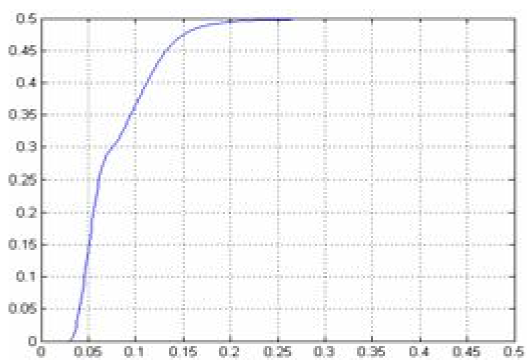


Figure 9: Step Response of the System

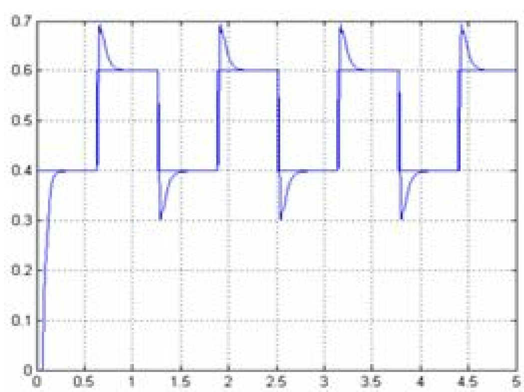


Figure 10: Square Wave Output Response of the System

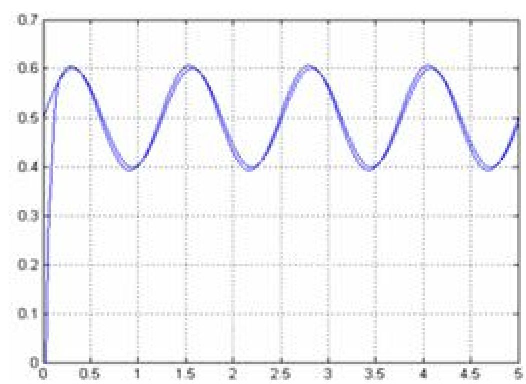


Figure 11: Sine Wave Output Response of the System

These two ICs have eight bits parallel interface with FPGA.

The fuzzy controller consists of three parts:

1- Summation part has two inputs set point and feedback, and generates the error and change of error that are inputs to fuzzy controller part.
2- Fuzzy controller is generated using Xfuzzy CAD tool and has two inputs (error and change of error) and one output (change of voltage).
3- PI (proportional-integral) part has one inputs (change of voltage) and one output (effective voltage).

The last step of Xfuzzy tool is generating the VHDL code for this controller; this code is used with “sum.vhd” and “pi.vhd” files to complete the controller. These files will be used with ISE 10.1 software to generate bit file that is uploaded to FPGA. Xfuzzy tool will generate System Generator blocks of fuzzy controller that is used with Matlab simulink. “Black Box” block that is in the System Generator Matlab toolbox will be used to insert the “sum.vhd” and “pi.vhd” into Matlab Simulink.

Figure 12 shows the overall system connected with all sub systems. After building the Simulink model of fuzzy controller using System Generator toolbox, the system runs

DOI: 01.IJEPE.4.2.1384

at two different set points, step, and sine wave to test the VHDL code if it works properly or not. Figure 13 shows the output response of the system at set point equal 127. (0.5 at normalize mode). Figure 14 shows the error and change of error of the ball position, the two values are shifted by 255, because the VHDL code of fuzzy controller works only in the interval [0 510], Where 0 means that the -255 and 510 means 255, so 0 error in Matlab equal 255 in VHDL code. We need to set the simulation time in System Generator block to be 0.00000002 sec to work as real FPGA because the FPGA clock that is used in VHDL code is 50 MHz

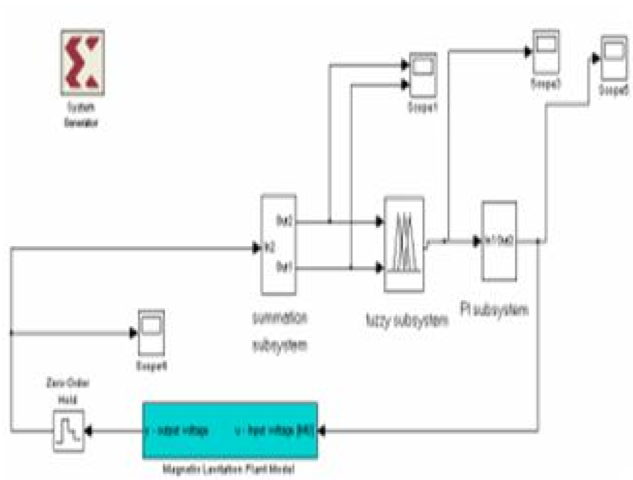


Figure 12: Block Diagram of Fuzzy Controller Using VHDL Code

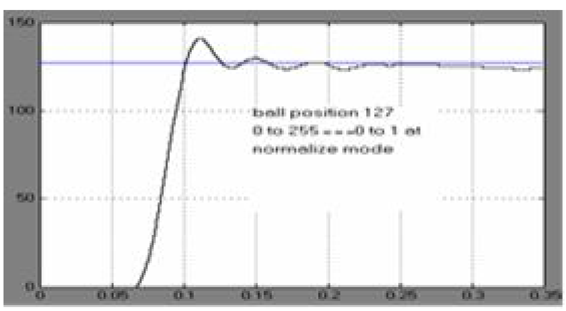


Figure 13: Step Response of the System Using VHDL Code

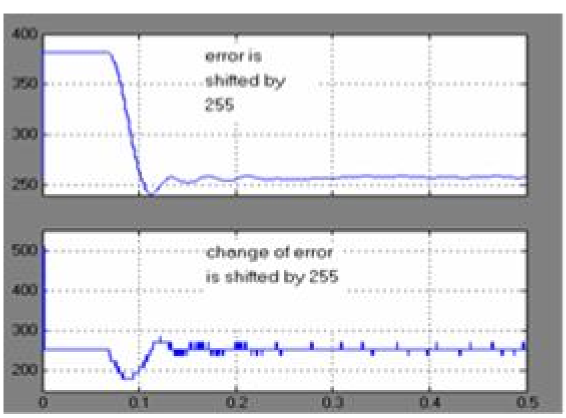


Figure 14: Error and Change of Error Using VHDL Code

Figure 15 shows the output response when the set point is sine wave.

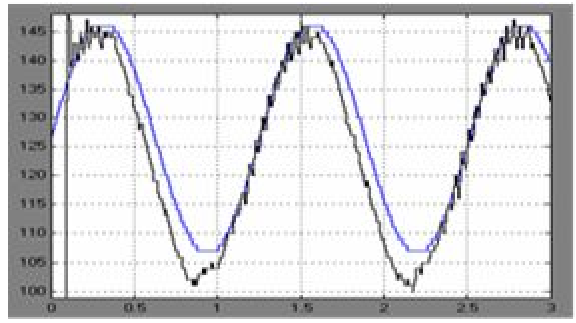


Figure 15: Sine Wave Output Response Using VHDL Code

C. GA Fuzzy Controller Design for CE152 Model

There are many parameters effect on the control process of the CE152 model as shown in Figure 8 a,b in addition to the shape of the memberships of the inputs and output of fuzzy controller. These parameters are p_g (proportional gain), pi_g (integral gain), ce_g (change of error gain), err_g (error gain) and out_g (fuzzy output gain). (See Figure 8 a,b and c) These gain parameters can be tuned to give near optimal results. GA is used to optimize these parameters and optimize the shape of memberships of fuzzy controller. The GA Matlab code uses next parameter

Population size= 50.

Number of gens in one chromosome = 13.

Crossover probability=0.9.

Mutation probability= random (0.0 to 1.0)

Stopping condition: 100 generations.

Figure 16 shows the change of the fitness values after running the Matlab code.

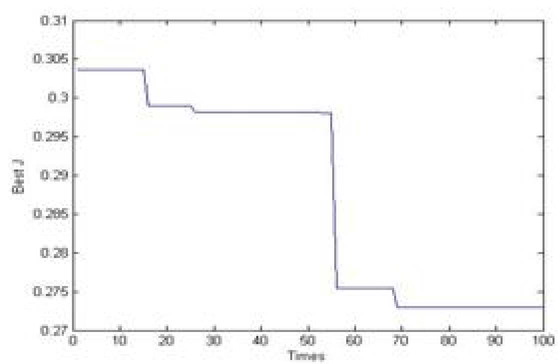


Figure 17 shows the memberships shape of the inputs and output

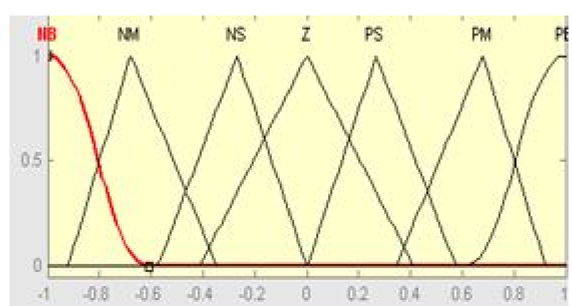


Figure 18: Membership Functions of GA Fuzzy Controller

DOI: 01.IJEPE.4.2.1384

Table 2 shows the difference between gain parameters with and without GA optimization

Table II. Gain Values with and without GA

Gains	with GA optimization	without GA optimization
err_g	1.0955	1
ce_g	15.4941	15
out_g	15.4932	15
p_g	0.15797	0.1
pi_g	2.7363	1

Figures 18, 19 and 20 show the system response of step, sin wave and square wave inputs with these new values.

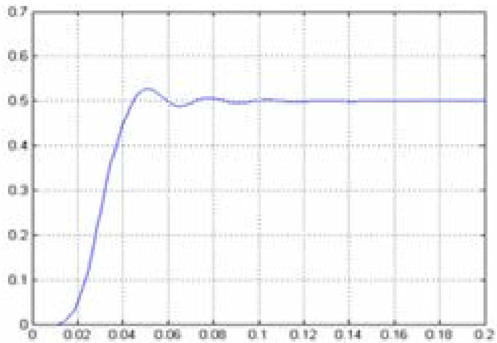


Figure 19: Step Response of the System with GA

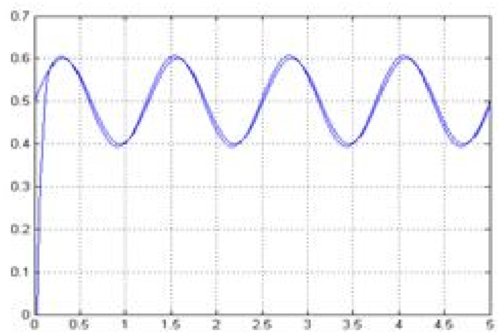


Figure 20: Sine Wave Output Response with GA

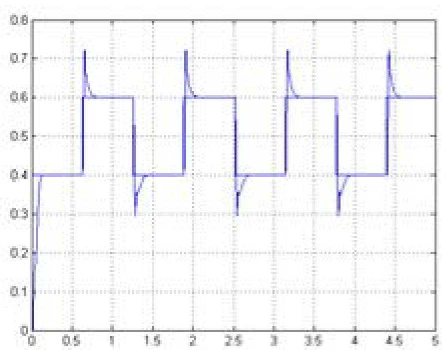


Figure 21: Square Wave Output Response with GA

Figure 18 shows that the overshoot of step input is nearly 6%, the settling time is 0.046 sec and rising time is 0.023 sec. Table 3 shows the comparison between the system response with and without GA optimization.

Table III. System Response with and without GA

System response for step input	With GA optimization	Without GA optimization
Overshoot	6%	No overshoot
Settling time	0.046 sec	0.2 sec
Rising time	0.023 sec	0.106 sec

As shown in Table 3 the Maglev response of fuzzy controller with GA optimization method is superior to fuzzy controller without GA.

VI. Conclusion

The real time implementation of the fuzzy logic controller for the various driving conditions and terrains has been achieved on a Xilinx Spartan 3E FPGA using VHDL. In this paper the magnetic levitation CE152 Model is used as practical example of nonlinear systems. The fuzzy controller was designed with Matlab software and this controller was tested with the CE152 Model. The fuzzy controller stabilized the magnetic levitation CE152 model under different set points. The Xfuzzy tool was used to generate the VHDL code of fuzzy controller, this code was used with "sum.vhd" and "pi.vhd" VHDL codes to give the overall controller of Magnetic Levitation CE152 Model. The GA optimization method was used to optimize the membership function of the inputs and output of the fuzzy controller and also to optimize the gains p_g, pi_g, ce_g, err_g and out_g. The CE152 was tested with the new membership functions and the result shows that is better than the results of old fuzzy controller under different set points.

References

[1] J. E. Bonilla, V.H. Grisales and M.A. Melgarejo "Genetic tuned FPGA based PD fuzzy LUT controller," IEEE International Conference. Fuzzy Systems, Vol 3,pp :1084 – 1087, 2001.

[2] O. Cordon, F. Herrera, F. Hoffman and L. Magdalena, "Genetic Fuzzy System Evolutionary Tuning and Learning of Fuzzy Knowledge Bases," Worlds Scientific, Singapore, 2001.

[3] Michael McKenna and Bogdan M. Wilamowski, "Implementing a Fuzzy System on a Field Programmable Gate Array," IEEE International Joint Conference on Neural Networks, Volume: 1, p.189-194, 2001.

[4] T. Philip, M. Vuong, asad, Madni and b. jim Vuong, . VHDL implementation for a fuzzy logic controller., BEI technologies, inc. 13100 telfair avenue, sylmar, 2006.

[5] S. Ravi, and P.A. Balakrishnan, "Stable Self Tuning Genetic Fuzzy Temperature Controller For Plastic Extrusion System," International Journal of Reviews in Computing, pp: 21-28 ,(2009-2011)

DOI: 01.IJEPE.4.2.1384

[6] Mohanad Alata, Mohammad Molhem and Khaled Al Masri, “Design of A Fuzzy Logic Controller for A Plant of N-Order Based on Genetic Algorithms,” International Conference on Robotics, Control and Manufacturing Technology, 2011

[7] John Yen & Langari Reza, “Fuzzy Logic Intelligence Control and Information,” Prentice-Hall, Englwood Cliffs, 1999.

[8] Maxfield, Clive, “The Design Warrior’s Guide to FPGAs,” Newnes, 2004

[9] Hosam Abu Elreesh, Basil Hamed,”FPGA Fuzzy Controller Design for Magnetic Ball Levitation”, IJISA, vol.4, no.10, pp.72-81, 2012.

[10] Robert Fulle´r, “Fuzzy Reasoning and Fuzzy Optimization,” TUCS General Publications, No. 9, Turku Centre for Computer Science, Abo, 1998.

[11] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, MA, 1989.

[12] J.H. Holland, Adaptation in Natural and Artificial systems, University of Michigan Press, Ann Arbor, 1975.

[13] S.N. Sivanandam and S.N. Deepa, Introduction to Genetic Algorithms, Springer, New York, , 2008.

[14] A. Alibeiki and S.S. Fallahi, “Genetic Algorithm and Comparison with Usual Optimization Methods,” World Applied Sciences journal 11 (6):752-754, 2010.

[15] Hung-Cheng Chen† and Sheng-Hsiung Chang, “Genetic Algorithms Based Optimization Design of a PID Controller for an Active Magnetic Bearing,” IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.12, December 2006.

[16] CE. 152 magnetic levitation model-education manual. Humusofts. r.o 2002.

DOI: 01.IJEPE.4.2.1384